IN THE SPECIFICATION

Please amend the specification as follows:

The paragraph beginning at page 7, line 4, is amended as follows:

Once an attempt to write to the protected area occurs, the method continues with checking the state of the write authorization flag ~~is checked~~ to determine whether writes to the protected area have been properly authorized (block 125). As mentioned above, the only program code with the ability to change the state of the write authorization flag resides in the protected area itself. Since this area can only be written when authorized, there is no way to modify the content of the area unless the approved code which resides in the protected area allows such access to occur. It is this self-validating mechanism which provides the essential security of the invention.

The paragraph beginning at page 7, line 17, is amended as follows:

If the write authorization flag is not set so as to authorize write operations to the protected area (as determined in block 125), then no further action need be taken. Carrying out the method at this point simply means that no action will be taken to enable write operation activity to the protected area. Even if write operations to the protected area proceed due to initiation from another part of the system, such as from a peripheral, no memory locations within the protected area will be changed, since the VPEN input has already been forced into a LOW logic state (see block 105). However, if it is determined in block 125 that the authorization flag has been set so as to permit writing to the protected area (typically using a set of software code instructions not located in the protected area to set the flag), then write operations, <u>including</u> ~~included~~ the attempted write operation, will be enabled (block 140). For the exemplary Intel 28F128J3 flash memory noted previously, this may be accomplished by either actively moving the input to the program/erase voltage switch (i.e., the VPEN input) to a HIGH state, or passively allowing the VPEN input to be moved to a HIGH state. In either case, one or more write operations may then proceed, until all writes to the protected area are completed (as determined in block 145). This can be accomplished, for example, by using the instructions resident within the protected area to trigger a particular type of interrupt after the desired number of write operations have been

completed. Until the interrupt is triggered, for example, write operations to the protected area are allowed to continue (block 145). However, once the interrupt is triggered, write operations are disabled (block 150). Again, to use the example of the Intel 28F128J3 flash memory, this may be accomplished by forcing the input to the program/erase voltage switch (i.e., the VPEN input) to a LOW logic state. At this point, the method continues with waiting to detect further attempts to write to the protected area (block 115).

The paragraph beginning at page 13, line 12, is amended as follows:

At this point, in block 330, write operations to the protected area are enabled. As mentioned above, in the exemplary case of a protected memory area located within an Intel 28F128J3 flash memory, this may be accomplished by either actively moving the input to the program/erase voltage switch (i.e., the VPEN input) to a HIGH state, or passively allowing the VPEN input to be moved to a HIGH state. In either case, one or more write operations may then proceed, until all writes to the protected area are completed, as determined in block 340, where for example, the completion of write operation activity can be detected by using the instructions resident within the protected area to trigger a particular type of interrupt, such as an [[a]] SMI interrupt. Until the SMI interrupt is triggered, write operations to the protected area are allowed to continue in block 340. However, once the SMI interrupt is triggered, write operations are disabled in block 345. Again, to use the example of the Intel 28F128J3 flash memory, this may be accomplished by preventing the input to the program/erase voltage switch (i.e., the VPEN input) from moving to a HIGH state. At this point, the method continues with block 350 by re-enabling the EXTSMI interrupt, and going on to block 310, waiting to detect further attempts to write to the protected area. Again, as will be appreciated by those skilled in the art, many other equivalent devices to the SMI interrupt (such as other interrupts, or hardware/software logic state changes) may be used to alert the processor to the completion of authorized write operations to the protected memory area.

The paragraph beginning at page 18, line 10, is amended as follows:

The memory access control circuit 400 also includes a memory 452 with a set of

instructions 457, such as an [[a]] SMM routine 457 in operational communication with the interrupt inputs 454 and 456. The instruction set 457 is adapted to activate an output indicating the state of the flag 481, which may be located in yet another memory 482, such as an unused non-volatile real-time clock register bit within the general purpose controller module 480. It should be noted that the memory 452 may be an integral part of the processor module 450 as shown in Figure 4, or optionally, the memory 451 with the second set of instructions 453 for determining the state of the flag 481 may be located externally, by using a separate DRAM 451 connected to the processor module 450 using the address and control lines 493, coupled to the processor module 450 using an Intel 82443ZX host bridge controller 490, for example. In any case, the memory 482 is in operational communication with the set of instructions 466, 468 located in the protected memory area 462.

The paragraph beginning at page 20, line 18, is amended as follows:

The processor module 550 is connected, directly or indirectly, to the access enabling line 575, and in turn, to the access enabling line input 563 of the memory 560. Such a connection may be effected for example, via address and control lines 593 which connect the processor module 550 to a general purpose controller module 580, which may be similar to or identical to an Intel 82371AB PIIX4 multifunction device. The access enabling line input 563 may be similar to or identical to the Intel 28F128J3 flash memory program/erase voltage switch input (i.e., the VPEN input), and may be controlled so as to enable or disable write operations to the protected area 562 by forcing the access enabling line 575 to a LOW logic state 584 using the instruction set 557 in the processor module 550 to activate the requisite address and control lines 593 to cause an output (such as the GPO9 output of the Intel 82371AB PIIX4 multifunction device) connected to the enabling access line 575 to go to a LOW logic state. As described previously, the processor module 550 includes an interrupt input 554 connected to the output of the write detection module 590. In order to detect completion of one or more authorized write operations to the protected memory area 562, the processor module 550 may include an interrupt, similar to or identical to a software SMI interrupt 556 which is operationally connected to the set of instructions 567 executed within the protected memory area. (i.e., [[ic.,]] software instructions,

hardware logic state transitions, or a combination of software and hardware may be used by the instruction set 567 to trigger the SMI interrupt 556).

The paragraph beginning at page 21, line 18, is amended as follows:

The memory access control circuit 500 also includes a memory 552 with a set of instructions 557, such as an [[a]] SMM routine, in operational communication with the interrupt input 554. The instruction set 557 is adapted to determine the state of the flag 581, which may be located in yet another memory 582, such as an unused non-volatile real-time clock register bit within the general purpose controller module 580. It should be noted that the memory 552 may be constructed as an integral part of the processor module 550 as shown in Figure 4, or optionally, the memory 552 may be located externally (not shown in Figure 5; see Figure 4), such as for a separate DRAM connected to the processor module 550 using the address and control lines 593 and any necessary bridge/control circuitry. In any case, the memory 582 is in operational communication with the set of instructions 567 located in the protected memory area 562.